Techniques for Securely Accelerating External Domains Locally

Hashem Mohammad Ebrahimi
Baber Amin
Mark D. Ackerman

Priority

The present invention is a continuation-in-part of commonly assigned and co-pending U.S. Application No. 10/784,440, filed on February 23, 2004 and entitled: "Techniques for Managing and Accelerating Data Delivery." U.S. Application No. 10/784,440 is a continuation-in-part to co-pending and commonly assigned U.S. Application No. 10/650,211 filed on August 28, 2003, entitled: "Secure Intranet Access," the disclosure of which is incorporated by reference herein. Pending U.S. Application No. 10/650,211 is a continuation of now issued U.S. Patent No. 6,640,302. Furthermore, U.S. Patent No. 6,640,302 is a divisional of now issued U.S. Patent No. 6,081,900.

Field of the Invention

The invention relates generally to network security, and more specifically to techniques for securely and locally accelerating data delivery associated with external domains.

Background of the Invention

A popular technique for accelerating external domains in a non-secure manner is through the use of site acceleration. Traditionally, to accelerate a complete domain, each site of a domain was associated with a reverse proxy which accelerated and managed data delivery for a particular site of that domain. A reverse proxy resides within the local environment of a site that is being accelerated.

In addition to reverse proxies, there are forward and transparent proxies which are used locally to cache and accelerate data to clients. A forward proxy is a service that a local client knows about and is configured to directly interact with when the client requests a resource of an external domain. A transparent proxy is a

service that the local client is unaware of and is unknowingly routed to when the local client requests a resource of the external domain.

Data acceleration occurs by the proxies maintaining caching data associated with the external domain within the local environment of the client. The client experiences improved data delivery and response time because data associated with needed resources or services of the external domain are housed locally within the proxies and vended to the client when needed. Thus, in many instances, the data needed by a client is locally available before the client requests that data and communication with a resource of the external domain is not needed at the time that the client makes the requests.

A domain is a logical collection of sites, resources, or services all identified by a common portion of an Internet Protocol (IP) address. For example, the site www.novell.com identifies a domain name of novell.com, which translates into a specific hierarchical portion of an IP address. The domain novell.com can be associated with a plurality of sites (services or resources), such as: sales.novell.com, support.novell.com, and access.novell.com. The services and resources can be further organized within specific sites associated with the domain.

Conventionally, to accelerate the entire domain of novell.com a reverse proxy is needed for each separate site of the domain. Recent technologies introduced by Novell, Inc. of Provo, Utah permit a single domain accelerator to be configured, such that all sites of a domain can be accelerated by a single service; rather than achieving acceleration via a plurality of separately configured reverse proxies.

Some domains are associated with secure communications, such as Hyper Text Transfer Protocol (HTTP) over Secure Sockets Layer (SSL) or Transport Layer Security (TLS), referred to as HTTPS. Secure communications are generally not cached in the local environments of clients, rather, the proxies establish a secure communications tunnel with the secure domains, sites, services, or resources and the proxies establish a secure communications tunnel with the clients. As a result, local data acceleration associated with secure communications has generally not been

available in the networking arts, since secure communications data is not locally cached.

However, recent techniques, such as the ones described in U.S. Application No., 10/784,440 entitled "Techniques for Managing and Accelerating Data Delivery" has changed this by providing techniques for securely accelerating remote sites within the local environments of clients.

Conventionally, external domains have been securely accelerated through the use of reverse proxies. A reverse proxy resides in the external domain and is external to the client and the client's environment. A reverse proxy provides more management benefits to the external domain than it provides acceleration benefits to the client. This is so, because the data contents of the reverse proxy's cache for the external domain are not cached within the local environments of requesting clients.

Thus, improved techniques for securely managing and accelerating external domains within local environments of clients are needed.

Summary of the Invention

In various embodiments of the invention, techniques are presented for securely managing and accelerating external domains within local environments of clients. External domains which securely communicate with clients using secure communications delegate their data delivery and management to local domain accelerators that reside in the local environments of the clients.

The local domain accelerators vend certificates and data on behalf of the external domains. That is, the local domain accelerator appears to the client to be an external domain. Communication between the local domain accelerator and the client occurs via a local secure communications channel using secure communications. Additionally, communication between the local domain accelerator and the external domain occurs over a secure communications channel using secure communications. The local domain accelerator acquires and caches data associated with sites, services or resources of the external domain. Thus, in many instances, when a client requests data from the external domain, that data is available within the cache of the local domain accelerator and can be vended to the

client. In this manner, data is securely accelerated to the client from within the local environment of the client.

Brief Description of the Drawings

FIG. 1 is a diagram representing an architectural layout for a secure external domain acceleration system;

FIG. 2 is a flowchart representing a method for securely accelerating an external domain locally;

FIG. 3 is a flowchart representing another method for securely accelerating an external domain locally; and

FIG. 4 is a diagram representing a secure external domain acceleration system.

Detailed Description of the Invention

As used herein and below a "client" is an electronic application, service, resource, or system which may be automated or may be manually interacted with by an end-user. Similarly, a proxy is a device, resource, service, system, or combinations thereof which act as an intermediary on behalf of clients as the clients interact with external domains. A proxy can be a forward proxy, which means the clients are configured to known about the proxy and configured to directly interact with the proxy. A proxy can also be transparent, which means the clients are not preconfigured to interact with the proxy, but some other service or device (*e.g.*, router, hub, bridge, switch, *etc.*) detects communications directed to or originating from the clients and directs them to the transparent proxy. In some embodiments, a single proxy can act as a forward proxy for some clients and as a transparent proxy for other clients.

An external domain is a local group of remote sites, services, or resources, which share some portion of an IP address with one another. That is, the remote sites, services, or resources have a common portion of their IP addresses which overlap. A remote site is a service, application, system, or resource with which the client desires to interact with in a secure manner for purposes of acquiring data or

information from that remote site. In order for the client to interact with a specific remote site, service, or resource, the client makes a request which has a site name or an IP address that identifies the external domain as well as the desired remote sites, services, or resources. In some embodiments, requests are made via SSL through a World-Wide Web (WWW) browser and identified within a Uniform Resource Locator (URL) or Uniform Resource Identifier (URI). The URLs or URIs include the site names or the IP addresses which identify the external domains.

The phrases "local networking environment" and "external (remote) networking environment are presented herein and below. Local networking environment refers to physical or logical network devices and services which are configured to be local to the clients and which interface with the clients. This does not mean that any particular local networking environment of a particular client physically resides in the same geographic location of the client or proximately resides within the same geographic location of the client, although in some embodiments this can be the case. Local networking environments can be dispersed geographically from the physical location of the client and form a logical local networking environment of the client. An external networking environment is a network which is not considered local to the client. An external domain is associated with external or remote networking environments, these external or remote networking environments are considered external and remote vis-à-vis a client's local networking environment.

Secure communications refer to communications that require specific secure protocols (e.g., HTTPS, SSL/TLS, *etc.*), which are communicated over predefined ports (*e.g.*, 443, *etc.*) associated with communication devices. In many cases data communication using secure communications requires encryption. In some instances this encryption uses Public and Private Key Infrastructure (PKI) techniques and which may also use digital certificates and digital signatures. Insecure communications refer to communications that use insecure protocols (*e.g.* HTTP, etc.) and which use different defined ports (*e.g.*, 80, *etc.*) of communication devices from that which are used with secure communications.

Data acceleration refers to the ability to cache data in advance of a need or request for that data. Any conventional caching services and managers can be used in the caching techniques presented herein and below with embodiments of this invention. Thus, by way of example, a cache manager can determine when to flush certain data from a cache and determine when certain data residing within the cache is stale and needs refreshed or updated. Generally, data is accelerated with caching techniques because the cache resides closer to a client and houses needed data in the local network of the client which is more quickly referenced and accessed. Thus, if a request for particular data can be satisfied from a local cache, a requesting client experiences a faster response time for that data and it appears to the client as if the data has been accelerated to satisfy a data request.

Various embodiments of this invention can be implemented in existing network products and services. For example, in some embodiments, the techniques presented herein are implemented in whole or in part in the iChain®, Border Manager®, and Excelerator® products distributed by Novell, Inc., of Provo, Utah.

Of course, the embodiments of the invention can be implemented in a variety of architectural platforms, systems, or applications. For example, portions of this invention can be implemented in whole or in part in any distributed architecture platform, operating systems, proxy services, or browser/client applications. Any particular architectural layout or implementation presented herein is provided for purposes of illustration and comprehension only and is not intended to limit the various aspects of the invention.

FIG. 1 is a diagram representing one example architectural layout 100 for a secure external domain acceleration system. The architecture 100 is implemented within a distributed client-server architecture. The purpose of the architecture 100 is to demonstrate how various entities can be configured and arranged for interacting, managing, and accelerating external domains over networks in local environments of requesting clients 101.

The architecture 100 includes a client 101, a proxy 102, optionally a switching device or logic 102A, and a local domain acceleration service 103. The client 101, the proxy 102, and the local domain acceleration service 103 are

configured within a physical or logical local networking environment with respect to one another.

The proxy 102 can be a transparent proxy, a forward proxy, or both a transparent proxy and forward proxy. Initially, the proxy 102 is configured by an administrator with a list of external domains 120 for which the proxy 102 creates a local domain accelerator 103 for each of the unique external domains 120 that are being locally accelerated. The proxy 102 handles secure requests received from the client 101 for a site, service, or resource of an external domain 120. If the proxy 102 is a transparent proxy the request from the client 101 is received via A and B through switch or router 102A. If the proxy 102 is a forward proxy the request from the client 101 is received directly via C.

The proxy 102 in response to a received request for an external domain 120 checks its list of valid domains for which a local domain accelerator 103 is available for locally accelerating the data associated with sites, services, and resources of the external domain 120. The proxy 102 listens for secure requests directed toward the external domain 120. When that secure request is detected, the proxy 102 establishes a secure channel between the requesting client 101 and the local domain accelerator 103.

In one embodiment, this is achieved by the proxy 102 using an anonymous SSL handshake to establish a secure channel between the client 101 and the proxy (*e.g.*, via C in FIG. 1). This is a temporary secure channel established to get the initial request. A host header of the initial request can be used to acquire the identity of the external domain 120 associated with the initial request. The external domain 120 houses the site, resource, or service for which the request desires. Further, the site, resource, or service has data which the request desires. Once the proxy 102 identifies the external domain 120, a new SSL handshake is requested so the appropriate domain certificate is returned. That domain certificate is associated with the identity of the needed external domain 120 and is acquired locally via the local domain accelerator 103. At this point, the anonymous connection is discontinued and switched to the local domain accelerator 103 via D, such that the

proxy 102 has established a local secure communications session between the requesting client 101 and the local domain accelerator 103.

The local domain accelerator 103 vends certificates on behalf of the external domain 120 and vends data on behalf of sites, services, and resources associated with the external domain 120. The local domain accelerator 103 communicates securely with the external domain 120 and its sites, service, and resources via D and E. The proxy 102 establishes a secure communications channel between the services, sites, and resources of the external domain 120 via E over a network 110. This establishes an external secure communications session between the local domain accelerator 103 and the external domain 120.

The local domain accelerator 103 presents itself and communicates with the requesting client 101 within the client's local networking environment via C-D or A-B-D in FIG. 1. The client 101 believes that it is directly communicating with an external domain 120 and one or more of its associated sites, services, or resources. This occurs because the local domain service 103 uses secure communications and is capable of vending certificates associated with the identity of the external domain 120.

The external domain 120 can have its certificate which it provides to the local domain accelerator signed by a Certificate Authority (CA), such that by providing the trusted root of the CA to the local domain accelerator 103, the established secure channel between the external domain 120 and the local domain accelerator 103 is made more secure. This is so, because no other authority could have signed the certificate sent by the external domain 120 during the SSL handshake. In some embodiments, all communications between the external domain 120 and the local domain accelerator 103 can be mutually signed in order to provide even more security with communications between the external domain 120 and the local domain accelerator 103.

Once a secure communication channel between the external domain 120 and the local domain accelerator 103 is established, the local domain accelerator acquires data from sites, resources, and services controlled by the external domain 120 and houses and manages that data in a cache that is within the local networking

environment of the client 101. In some embodiments, the cache and caching services are provided by the proxy 102 to the local domain accelerator for purposes of managing and accelerating the delivery of that data. Thus, the entire external domain 120 can be managed and locally accelerated by the local domain accelerator 103.

The local domain accelerator 103 can vend the certificate and data associated with the external domain 120 in a variety of manners. For example, the local domain accelerator 103 can house the data associated with sites, services, or resources of the external domain 120 in a decrypted format. In this scenario, a client that requests data from the external domain 120 is provided the desired data from the local cache in an encrypted format that is encrypted with the SSL session keys (session keys used between the client 101 and the local domain accelerator 103) along with a certificate identifying the local domain accelerator 103 as the external domain 120.

In other embodiments, the data may be housed within the local cache in an encrypted format, which is decrypted when needed, and then re-encrypted with the SSL session key of the client 101 and the local domain accelerator 103 and delivered in the re-encrypted format to the client 101 along with a certificate that presents the local domain accelerator 103 as the external domain 120. Thus, the local domain accelerator 103 can natively house data it receives from the external domain 120 in decrypted or encrypted formats and can vend that data to the client 101 in encrypted formats expected by the local secure communications session along with a certificate that indicates to the client that the communication is coming from the external domain 120.

The client 101 believes that it is securely communicating with specific resources, services, or sites of an external domain 120 and the client 101 experiences accelerated data delivery. Data delivery is accelerated because the local domain accelerator 103 acquires or pre-fetches desired data and vends it from the local cache. In this manner, an external domain 120 is locally accelerated in a secure fashion to the client 101.

The architecture 100 of FIG. 1 can be achieved with minimal changes to existing networking architectures. Existing proxies are configured to inspect secure communication requests in order to identify the targeted external domains 120. If a target external domain 120 is associated with a local domain accelerator 103, then the request for secure communications is passed from the proxy 102 to the identified local domain accelerator 103. The local domain accelerator 103 translates and manages secure communications received from the sites, services, or resources of the external domain 120 into secure communications expected by the client 101.

FIG. 2 is a flowchart of one method 200 for securely accelerating an external domain over a network and within a local networking environment of a client. The method 200 (hereinafter "processing") is implemented in a computer readable and accessible medium. In one embodiment, the processing represents services provided by a proxy 102 and a local domain accelerator 103, such as the services and processing discussed above with respect to the architecture 100 of FIG. 1.

Initially, an architectural layout similar to the architecture 100 of FIG. 1 is configured and arranged. This configures a local networking environment for a client, such that a local domain accelerator is in a position to locally vend data and certificates securely to the client on behalf of an external domain. That data is vended to the client securely (*e.g.,* using SSL, TLS, *etc.*) from a local cache.

At 210, the processing receives securely (via secure communication protocols) an initial request for a site, service, or resource of an external domain. That initial request can be received directly from the client in cases where the processing is acting as a forward proxy. Alternatively, the initial request can be received indirectly from the client via a router, switch, hub, or bridge in cases where the processing is acting as a transparent proxy. Upon receipt of the initial request the processing establishes a temporary secure communications channel with the client, such as through an anonymous SSL handshake.

At 220, the processing inspects a portion of the request for purposes of acquiring and identifying an external domain name (identifier) associated with the request. When the request is an URL or URI one technique for acquiring the domain name is by stripping from the URL or URI the host header at 221, which

provides the domain identification information to the processing. Next, the processing determines if the acquired domain identifier is listed in a lookup table or other data structure as one in which secure communications are being managed and accelerated locally by a local domain accelerator, such as the local domain accelerator 103 described above with respect to the architecture 100 of FIG. 1.

At this point in time, the processing routes the initial client request for a site, resource, or service of the identified external domain to the local domain accelerator at 230. The anonymous and temporary secure communications between the proxy and the client can then be made into an active secure communications session. One technique for achieving this is to establish an SSL or TLS session at 231 where session keys are established between the client and the local domain accelerator for purposes of encrypting and decrypting communications. Furthermore, at 232, the local domain accelerator presents a certificate to the client that identifies the local domain accelerator as the external domain. Now, the client and local domain accelerator have a local secure communications session and the client believes that it is directly interacting with the external domain and its associated sites, services, or resources.

Correspondingly, at 233, the local domain accelerator, which resides in the local networking environment of the client, services the client from local cache during the secure communications session with data acquired from sites, resources, or services of the external domain. In some embodiments, at 234, the local domain accelerator utilizes the caching services associated with the processing (*e.g.,* proxy).

The local domain accelerator populates, flushes, and pre-fetches data from selective sites, services, and resources of the external domain into the local cache. Communications between the local domain accelerator and the external domain occur securely via a secure communications channel during an external secure communications session. In some embodiments, the communications are mutually signed or signed by only one of the parties. The external secure session between the local domain accelerator and the external domain encrypts data using its unique session key, and each party decrypts data using their unique decrypting session key. Encryption and decryption typically takes place using public-private key pairs, such

that decryption of the local domain accelerator occurs with a public key of the external domain and a private key of the local domain accelerator. Similarly, decryption of the external domain occurs with a public key of the local domain accelerator and a private key of the external domain.

When the local domain accelerator vends data during its local secure session with the client from the local cache, the data which was pre-acquired from the external domain is translated into an encrypted format that is being used with that local secure session in manners similar to what was discussed above. The difference is that the session keys for the local secure session between the client and the local domain accelerator are different from the session keys of the external secure session being used between the local domain accelerator and the external domain. The local domain accelerator also presents a certificate to the client which informs the client that the local domain accelerator is the external domain.

The techniques of the method 200 permit entire external domains or collections of external domains to be locally accelerated and vended within the local networking environments of clients. Thus, data management responsibilities of external domains can be delegated to trusted local domain accelerators and data acceleration occurs within the local networking environments of the clients. As a result, clients experience improved data delivery for sites, resources, and services associated with external domains while still maintaining and using traditional secure communications to acquire that data.

FIG. 3 is a flowchart of another method 300 for securely accelerating the delivery of data associated with external domains within the local networking environments of requesting clients. The method 300 (hereinafter "processing") is implemented in a computer readable and accessible medium within the local networking environment of a client, where the client, and a needed service, site, or resources of an external domain, desire to interact securely with one another. In one embodiment, the processing reflects the services or operations which are performed by a proxy 102 and a local domain accelerator 103 associated with the architecture 100 of FIG. 1.

The processing can be a forward proxy, transparent proxy, or both a forward proxy and transparent proxy. Initially, at 310, the processing (proxy) receives a secure communications request for a specific site, resource, or service of a specific external domain. This initial contact with the processing creates a temporary and anonymous SSL handshake which is subsequently converted into a more permanent SSL handshake when the processing contacts a needed local domain accelerator that can vend to the client a certificate of the external domain. That certificate identifies the local domain accelerator as the external domain as far as the client is concerned and permits the establishment of a local secure communications session with the client at 320.

At 310A, concurrent with, prior to, or after the processing facilitates the local secure communications session between local domain accelerator, an external secure communications session is established via the processing between the external domain and the local domain accelerator. In some embodiments, at 311A, communication during the external secure communications session can be mutually signed for added security or unilaterally signed by at least one of the parties. Of course, in other embodiments, no signing of communications is needed at all during the external communications session.

During the external secure communications session, the local domain accelerator acquires data from selective or all sites, services, and resources controlled by the external domain and populate and manage that data in a local cache at 320A. In some embodiments, the local cache and caching services are existing and conventional techniques that are provided by the processing to the local domain accelerator. Data sent and received during the external secure communications session is encrypted and decrypted using the external secure sessions' keys.

In one embodiment, at 321, the processing (proxy) facilitates the creation of the external secure communications session by acting as a conduit between the local domain accelerator and the external domain and by creating a secure communications channel for the external secure session to proceed.

At 322, the local domain accelerator also maintains a separate and unique local secure communications session with the client during which a certificate of the external domain is vended or presented to the client that identifies the local domain accelerator as the external domain as far as the client is concerned. During the local secure communications session data associated with services, resources, or sites of the external domain is encrypted and decrypted between the local domain accelerator and the client using their unique local secure sessions' keys. In this way, the local domain accelerator translates encryption of the external secure communications session into encryption understood and needed by the client for the local secure communications session.

Once the two secure communications sessions are established, such that the local domain accelerator is the only party participating in both sessions, the local domain accelerator can service the client and its requests for data from the local cache at 330. In this way, the local domain accelerator acts as a secure external domain accelerator on behalf of services, resources, and sites associated with the external domain. In a sense, the local domain accelerator becomes a local reverse proxy for the external domain. A single external domain can delegate and interact with multiple local domain accelerators where each local domain accelerator is associated with accelerating data on behalf of the external domain in different client domains.

FIG. 4 is diagram representing a secure external domain acceleration system 400. The secure external domain acceleration delivery system 400 is implemented in a computer readable and accessible medium and operates over one or more networks. The networks can be hardwired, wireless, or a combination of hardwired and wireless. In one embodiment, various processing aspects which were described above with respect to the methods 200 and 300 of FIGS. 2 and 3, respectively, are implemented within the secure external domain acceleration system 400.

The secure external domain acceleration system 400 includes a local domain accelerator 401, a cache 402, and a proxy 403. In some embodiments, the cache 402 is integrated into or forms a logical part of the proxy 403. In other embodiments, the local domain accelerator 401 can be integrated into or form a

logical part of the proxy 403 as well. Moreover, in one embodiment, the cache 402 is uniquely associated with or integrated with the local domain accelerator.

The proxy 403 can be a forward proxy or a transparent proxy. Alternatively, the proxy can be designated as either a forward proxy or a transparent proxy based on how it interacts with a specific client 410. In this way, the proxy 403 need not exclusively be either a forward or transparent proxy, but can in some arrangements be both a forward proxy and a transparent proxy.

The proxy 403 facilitates and acts as a conduit for establishing a local secure communications session between the local domain accelerator 401 and one or more local clients 410. Additionally, the proxy 403 facilitates and acts as a conduit for establishing an external secure communications session between the local domain accelerator 401 and an external domain 420. The local domain accelerator 401 is the only participant which interacts with both secure sessions and in this manner acts as a translator between data being migrated between different sessions.

The external secure communications session can be established before the local secure communications session, established concurrent with the local secure communications session, or established after the local secure communications session. Delivery of data associated with sites, services, or resources of the external domain 420 occurs during the local secure communications session and is vended by the local domain accelerator 401 from the cache 402. That is, the local domain accelerator 401 uses its external secure communications session to pre-acquire and pre-fetch needed data into the cache 402, such that when a local client 410 requests that data during the local secure communications session the local domain accelerator 401 is in a position to translate that data from the cache into an encrypted format needed by the local session and delivery it to the requesting client 401.

Thus, the secure external domain acceleration system 400 processes within the local networking environment of clients 410 for purposes of locally accelerating data delivery to those clients 410 during a locally secure communications session. As a result, clients 410 can now experience accelerated data delivery for entire external domains 420 within their local networking environments. This improves

the throughput of data delivery associated with sites, services, and resources of external domains 420 and offloads processing that would normally take place within the networking environments of the external domains 420.

Although specific embodiments have been illustrated and described herein, those of ordinary skill in the art will appreciate that any arrangement calculated to achieve the same purpose can be substituted for the specific embodiments shown. This disclosure is intended to cover all adaptations or variations of various embodiments of the invention. It is to be understood that the above description has been made in an illustrative fashion only. Combinations of the above embodiments, and other embodiments not specifically described herein will be apparent to one of ordinary skill in the art upon reviewing the above description. The scope of various embodiments of the invention includes any other applications in which the above structures and methods are used. Therefore, the scope of various embodiments of the invention should be determined with reference to the appended claims, along with the full range of equivalents to which such claims are entitled.

It is emphasized that the Abstract is provided to comply with 37 C.F.R. §1.72(b), which requires an Abstract that will allow the reader to quickly ascertain the nature and gist of the technical disclosure. It is submitted with the understanding that it will not be used to interpret or limit the scope or meaning of the claims.

In the foregoing Detailed Description, various features are grouped together in single embodiments for the purpose of description. This method of disclosure is not to be interpreted as reflecting an intention that the claimed embodiments of the invention require more features than are expressly recited in each claim. Rather, as the following claims reflect, inventive subject matter lies in less than all features of a single disclosed embodiment. The following claims are hereby incorporated into the Detailed Description, with each claim standing on its own as a separate preferred embodiment.